Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

Кафедра комп’ютерних та інформаційних технологій і систем

(повна назва кафедри)

**Пояснювальна записка**

до дипломної роботи

бакалавра

(ступінь вищої освіти)

на тему:

**«**Електронний довідник автоматизованого обладнання засобами web-технологій**»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Виконав: студент | | 4 | курсу, групи | 402-ТК |
| напряму підготовки (спеціальності) | | | | |
| 123 Комп’ютерна інженерія | | | | |
| (шифр і назва напряму підготовки, спеціальності) | | | | |
| Кривошлик В.С. | | | | |
| (прізвище та ініціали) | | | | |
| Керівник | к.т.н. Мавріна М.О. | | | |
|  | (прізвище та ініціали) | | | |
| Рецензент |  | | | |
|  | (прізвище та ініціали) | | | |

м. Полтава – 2021

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**спеціальність 123 «Комп’ютерна інженерія»**

**на тему**

**«Електронний довідник автоматизованого обладнання засобами web-технологій»**

**Студента групи 402-ТК Кривошлика Валентина Сергійович**

Керівник роботи

ст. викладач

Мавріна М.О.

Завідувач кафедри

к.т.н. доц.

Головко Г. В.

Полтава – 2021

**РЕФЕРАТ**

Загальний обсяг текстової частини дипломної роботи складає 52 с., формату А4. Вона складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаних джерел та одного додатку. Робота містить 12 рисунків, використано 11 науково-технічних джерел.

Дипломна робота бакалавра присвячена створенню електронного довідника автоматизованого обладнання.

У першому розділі дипломної роботи розглядаються теоретичні основи розробки веб-сайтів.

У другому розділі розглядаються методи та засоби веб-технологій.

У третьому розділі наведена поетапна розробка веб-ресурсу засобами HTML, CSS, PHP. Описано технології, які використовувались для розробки та приведені приклади функціонування веб-ресурсу, можливості взаємодії з користувачем.

Мета дослідження – створення web-сайту автоматизованого обладнання.

Об’єкт дослідження – процес розробки веб-сайтів.

Предмет дослідження – засоби розробки веб-сайтів.

Метод дослідження – аналітичний із використанням комп’ютерних технологій.

Ключові слова: HTML, CSS, PHP,JS, WWW, веб-сайт, веб-сервер

**ABSTRACT**

The total volume of the text part of the thesis is 52 A4 pages. It consists of a list of abbreviations, an introduction, three sections, conclusions, a list of sources used and one appendix. The work contains 12 drawings, 11 scientific and technical sources were used.

The bachelor's thesis is devoted to the creation of an electronic directory of automated equipment.

The first section of the thesis discusses the theoretical foundations of website development.

The second section discusses the methods and tools of web technologies.

The third section presents the step-by-step development of a web resource using HTML, CSS, PHP. The technologies used for development are described and examples of functioning of a web resource, possibilities of interaction with the user are resulted.

The purpose of the study is to create a web-site of automated equipment.

The object of research is the process of website development.

The subject of research - the tools of website development.

Research method - analytical using computer technology.

Keywords: HTML, CSS, PHP,JS, WWW, web-site, web-server

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 6

ВСТУП 7

РОЗДІЛ 1 ТеоретичнІ засади створення веб­Сайтів 8

1.1Веб-сайт. 9

1.2.Класифікація сайтів 10

1.3Структура веб-сайту 11

1.4. Етапи розробки веб-сайту 12

1.6Висновки за розділом 1 15

РОЗДІЛ 2 Методи та засоби WEB-ТЕХНОЛОГІЙ 16

2.1 HTML 16

2.2 CSS 19

2.3PHP 24

2.4 JS 29

2.5 Висновки за розділом ……………………………………………….32

РОЗДІЛ 3. ПРОЕКТНА ЧАСТИНА 33

3.1. Розробка інтерфейсу веб-сайту 33

3.2.Реєстрація сайту в мережі Інтернет 36

3.3 .Тестування веб-сайту в різних браузерах………………………….38

3.4. Висновки до розділу 3…………………………………………........41

ВИСНОВКИ 42

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43

ДОДАТОК А. ЛІСТИНГ КОДУ ДОДАТКУ 45

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

|  |  |  |
| --- | --- | --- |
| **HTML** | – | Мова розмітки гіпертексту |
| **CSS** | – | Каскадні таблиці стилів |
| **PHP** | – | Скриптова мова програмування |
| **XHTML** | – | Розширювана мова розмітки гіпертексту |
| **URL** | – | Cтандартизована адреса певного ресурсу |
| **JS** | – | Мова програмування |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**ВСТУП**

Інформаційні технології набувають все більшої актуальності та мають вплив на суспільний розвиток, процеси інформатизації зачіпають всі сфери діяльності людини. Еволюція сфери інформаційних технологій є чинником, що впливає на зміни в економіці, політиці, культурі та освіті. У цих умовах інформація стає одним з основних стратегічних ресурсів, а інформаційні технології - інструментом, з допомогою якого цей ресурс використовується. Процеси суспільного розвитку призводять до поступової зміни індустріального суспільства інформаційним. Інформаційні технології набувають все більшої актуальності та мають вплив на суспільний розвиток, процеси інформатизації зачіпають всі сфери діяльності людини. У цих умовах інформація стає одним з основних стратегічних ресурсів, а інформаційні технології - інструментом, з допомогою якого цей ресурс використовується.

Метою дипломної роботи є створення веб сайту за допомогою новітніх технологій.

**Розділ 1.**

**ТеоретичнІ засади створення веб-Сайтів**

Інтернет (від англ. Internet) – всесвітня система добровільно об'єднаних комп'ютерних мереж, побудована на використанні стеку протоколів TCP/IP і маршрутизації пакетів даних. [1]

Інтернет мережу утворюють мільйони веб-серверів мережі Інтернет, які знаходяться по всьому світу. Веб-сервер це програма, що запускається на підключеному до мережі комп'ютері і використовує протокол HTTP для передачі даних. У звичайному вигляді ця програма отримує по мережі HTTP-запит на певний ресурс, знаходить файл на локальному жорсткому диску і доставляє його по мережі комп'ютеру. Складніші веб-сервери здатні динамічно формувати ресурси у відповідь на HTTP-запит. [1].

Для ідентифікації ресурсів у всесвітній мережі використовуються одноманітні ідентифікатори ресурсів URI. Для знаходження місцезнаходження ресурсів в мережі використовуються одноманітні локатори ресурсів URL (англ. Uniform Resource Locator). Такі URL-локатори поєднують в собі технологію ідентифікації URI і систему доменних імен DNS (англ. Domain Name System) – доменне ім'я (або безпосередньо IP-адрес в числовому записі) входить до складу URL для позначення комп'ютера (точніше – одного з його мережевих інтерфейсів), який виконує код потрібного веб-сервера. [1].

Для перегляду інформації, отриманої від веб-сервера, на клієнтському комп'ютері застосовується спеціальна програма – веб-браузер. Основна функція веб-браузера – відображення гіпертексту. [1].

В цілому можна зробити висновок, що Всесвітня мережа стоїть на «трьох китах»: HTTP, HTML і URL. Хоча останнім часом HTML почав трохи здавати свої позиції і поступатися ними сучаснішим технологіям розмітки: XHTML і XML [1].

* 1. **Веб-сайт**

Сайт (від англ. website: web – «павутина, мережа» і site – «місце», буквально «місце, частина в мережі») –поєднання електронних файлів приватної особи або організації в комп'ютерній мережі, об'єднаних під однією адресою [2].

Всі сайти складають Всесвітню павутину, де комунікація (павутина) об'єднує сегменти інформації світової спільноти в одне ціле – базу даних і комунікації величезного масштабу. Для прямого доступу клієнтів до сайтів на серверах був спеціально розроблений протокол HTTP[2].

Сторінки сайтів – це набір текстових файлів, написаних мовою HTML. Ці файли, будучи завантаженими відвідувачем на його комп'ютер, розуміються і обробляються браузером і виводяться на засіб відображення користувача (монітор, екран КПК, принтер або синтезатор мови). Мова HTML дозволяє форматувати текст, розрізняти в ньому функціональні елементи, створювати гіпертекстові посилання (гіперпосилання) і вставляти в сторінку зображення, звукозаписи і інші мультимедійні елементи. Відображення сторінки можна змінити додаванням до неї таблиці стилів CSS, що дозволяє централізувати в певному файлі всі елементи форматування або сценарії на мові JavaScript, за допомогою яких є можливість переглядати сторінки з подіями або діями[2].

Сторінки сайтів можуть бути простим статичним набором файлів або створюватися спеціальною комп'ютерною програмою на сервері. Вона може бути, зроблена на замовлення для окремого сайту, або бути готовим продуктом, розрахованим на певний клас сайтів. Деякі з них можуть забезпечити власнику сайту можливість гнучкої настройки структуризації і виведення інформації на веб-сайті. Такі керуючі програми називаються системами управління вмістом (CMS) [2].

* 1. **Класифікація сайтів**

За доступністю сервісів їх поділяють на:

Відкриті – усі сервіси доступні для кожного користувача;

Напіввідкриті – для доступу потрібно зареєструватися, або створити особистий кабінет;

Закриті – повністю закриті сайти деяких організацій (корпоративні сайти, сайти для деяких груп населення, особисті сайти приватних осіб). Такі сайти можуть бути доступні для невеликої кількості людей. Зазвичай доступ до сайту можливий через запрошення[3].

За фізичним розташуванням їх поділяють на: якщо сайт доступний користувачам з Інтернету, він вважається зовнішнім, проте сайт, доступ до якого можуть отримати лише користувачі локальної мережі, є внутрішнім. Такими сайтами можуть бути корпоративні сайти підприємства або особистий сайт в локальній мережі провайдера[3].

За призначенням сайти поділяють на:

Бізнес-сайти – сайти, що містять інформацію про компанії та їхні послуги, мають можливість торгівлі;

Інформаційні сайти – призначені для викладання інформації, поширення новин

Сайти соціальних мереж – інтерактивні багатокористувацькі веб-сайти, які поповнюються самими учасниками мережі. Сайт являє собою автоматизоване соціальне середовище, що дозволяє спілкуватися групі користувачів, об’єднаних спільним інтересом; [3].

Веб-портали – сайти, через які можна знайти все, що тобі потрібно в мережі інтернет;

Сайти сервісів – сайти служб, які існують у мережі Інтернет, зокрема, сайти пошукових служб (Google, Yahoo, Ukrnet), поштові сайти(Mail,Ukr.net), веб-форуми, онлай диски (Google диск), зберігання відео (You Tube).

* 1. **Структура веб-сайтів**

Зовнішній вигляд кожного сайту є унікальним, однак майже всі сайти хоч і мінімально, але все ж схожі між собою. На кожному сайті першою сторінкою є головна сторінка, її розробці приділяють особливу увагу, так як люди не читають всю інформацію на моніторі. Зазвичай вони переглядають її не дуже уважно. Якщо головна сторінка містить саме ту інформацію, що шукає людина, то користувач залишиться на сайті, а якщо ні – переходить до інших сайтів, яких в Інтернеті дуже багато. У верхній частині головної сторінки завжди розташована так звана шапка, яку дублюють на інших сторінках сайту. Це роблять оскільки, вона першою привертає увагу користувача після відкриття сайту. Задля швидкого переходу по сайту, створюють меню сайту – список гіперпосилань на його розділи. Зазвичай використовують два меню: вертикальне і горизонтальне. Меню є дуже важливим компонентом сайту, користувач завжди звертає на нього увагу, і тому до його створення підходять з відповідальністю. Меню має бути легким і зрозумілим, інакше користувач може не знайти перехід до потрібного розділу, і залишить сайт. Пункти меню повинні бути відділені один від одного[4].

Гіперпосилання, розміщені в тексті або у вигляді графічних об'єктів, дозволяють переходити на всі сторінки сайту або навіть на інші сайти. На сайтах із дуже великим обсягом інформації є сторінки різних рівнів. Взагалі виділяють три типи структур веб-сайтів – лінійну, деревоподібну та довільну.

Відвідуючи сайт із лінійною структурою, з головної сторінки користувач буде переходити на інші сторінки послідовно[4].

На сайті з деревоподібною структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти – на одну зі сторінок третього рівня. Сайт із довільною структурою дозволяє переходити на сторінки різних рівнів у довільному порядку. Вибір структури залежить від виду сайту і його завдання[4].

* 1. **Етапи розробки веб-сайту**

Створення сайту можна розділити на декілька етапів:

1. Попередній етап розробки веб-сайту: з’ясовується загальна концепція сайту, вирішуються завдання і цілі створення сайту.
2. Етап проектування структури сайту: меню, розміщення модулів, тощо. Результатом цього етапу є створення документів, що описують задачі сайту: розміщення елементів дизайну, структуру системи навігації, особливості роботи всіх модулів.
3. Етап розробки й тестування сайту.

Тестування програмного забезпечення на відсутність помилок, правильність обчислень, стійкість до перевантажень та юзабіліті-тестування.

1. Розміщення сайту:

* на локальному сервері установи;
* на сервері вашого провайдера;
* на сервері організації, яка спеціалізується у наданні послуг розміщення сайтів користувачам Інтернету;
* на сервері, який надає послуги безкоштовного розміщення сайтів.

1. Розвиток ресурсу.

Після розміщення сайту в Інтернеті потрібно здійснювати його підтримку, щоб сайт не втрачав своєї популярності. Ця підтримка полягає в періодичному оновленні та доповнені існуючих матеріалів, створенні нових цікавих сторінок, тощо[5].

Оскільки, процес розробки є складним його можна розділити на клієнтську (front-end) та серверну (back-end) частини розробки.

Front-end – це створення користувальницького інтерфейсу, функціональності та інтерактивності, що виконується на стороні клієнта. Щоб зрозуміти, що таке front-end розробка, необхідно відкрити сторінку будь-якого ресурсу та переглянути код даної сторінки в браузері. Цей код є прикладом роботи front-end розробки, що завантажується в браузер користувача, де відбувається його парсинг. Код сторінки описує все, що користувач бачить перед собою: кольори, верстку, шрифти, розташування графічних елементів тощо[5].

Сучасна front-end розробка тримається на трьох основних мовах: HTML, CSS, JavaScript, де HTML відповідає за структуру, CSS за дизайн, а JavaScript за інтерактивність (рис. 1.1).

Щоб веб-ресурс був більш досконалим, front-end розробник співпрацює з дизайнерами, програмістами та UX-аналітиками, для створення зручного та конкурентоспроможного продукту.

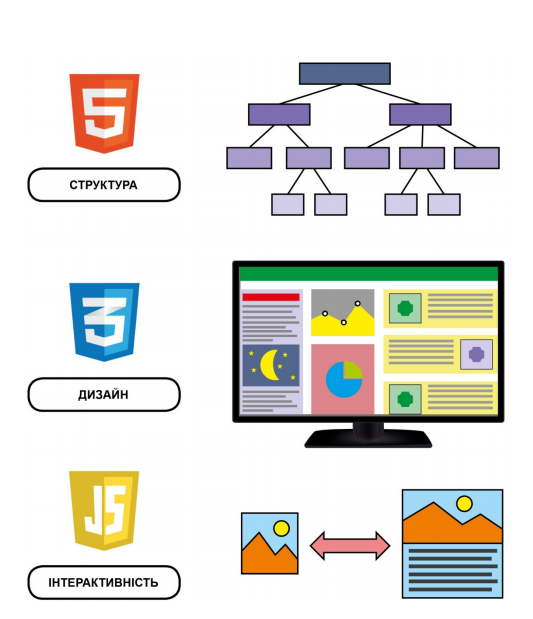


Рисунок 1.1 Організація front-end розробки

Back-end розробка – це набір апаратно-програмних засобів, за допомогою яких реалізується логіка роботи ресурсу. Тобто, це те, що приховано від очей користувача і відбувається поза його браузером та комп’ютером. Яскравий приклад back-end розробки: коли користувач вводить запит на сторінці браузера та натискає клавішу Enter. Запит відправляється на сервер, де розташовані алгоритми пошуку. Сервер – це більш потужний комп’ютер, що виконує певні функції. Він зберігає дані та відповідає на запити користувачів. Щойно на моніторі з’являється інформація, яку шукав користувач, знову відбувається повернення в зону front-end. По суті, back-end – це процес об’єднання сервера з користувачем за допомогою баз даних, API та операційних систем[5].

Back-end розробник може застосовувати різноманітні інструменти, що доступні на сервері. Як правило, це мови програмування, зокрема PHP, Java та Python. Також для back-end розробки використовуються різні системи управління базами даних: MySQL, MongoDB, Cassandra тощо. Самою поширеною комбінацією в back-end розробці є поєднання PHP з MySQL (рис. 1.2). Залежно від цілей, back-end розробник може виконувати наступні функції: забезпечувати кібербезпеку ресурсу та даних користувачів, створювати та інтегрувати бази даних, налаштовувати технології резервного копіювання та відновлення[5].

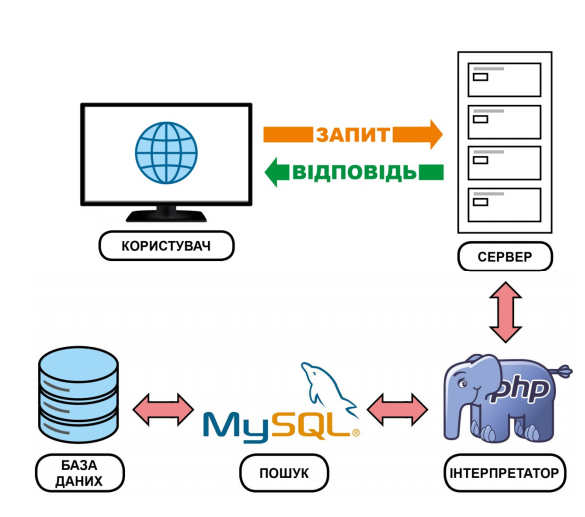


Рисунок 1.2 – Організація back-end розробки

**1.5 Висновок за розділом 1**

Глобальна мережа складається з усіх сайтів, доступних для загального користування. В розробці сайтів бере участь велика кількість людей - програмісти, веб-дизайнери та інші.

В першому розділі було розглянуто теоретичні засади створення веб-сайтів, а також я визначив етапи створення веб-сайту.

**Розділ 2. Методи та засоби WEB-ТЕХНОЛОГІЙ**

### **2.1 HTML**

Мова розмітки HyperText або HTML - це стандартна мова розмітки для документів, призначених для відображення у веб-браузері. Йому можуть допомогти такі технології, як каскадні таблиці стилів (CSS) та мови сценаріїв, такі як JavaScript. [6].

Веб-браузери отримують документи HTML з веб-сервера або з локальної пам’яті та надають документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та включає ознаки зовнішнього вигляду документа. [6].

Елементи HTML - це блоки HTML-сторінок. За допомогою конструкцій HTML зображення та інші об’єкти, такі як інтерактивні форми, можуть бути вбудовані у створену сторінку. HTML забезпечує засіб для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML розмежовані тегами, записаними за допомогою кутових дужок. Такі теги, як <img /> та <input />, безпосередньо вносять вміст на сторінку. Інші теги, такі як <p>, оточують та надають інформацію про текст документа і можуть включати інші теги як піделементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки. [6].

HTML може вбудовувати програми, написані мовою сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд та макет вмісту. Консорціум World Wide Web (WWW), колишній супровідник HTML і поточний супровід стандартів CSS, заохочує використання CSS над явним презентаційним HTML з 1997 року. [6].

Різниця між версіями HTML:

Хронологія версій HTML

**HTML 2**

24 листопада 1995 р

HTML 2.0 був опублікований як RFC 1866. Додаткові RFC додали можливості:

1)25 листопада 1995 р.: RFC 1867 (завантаження файлів на основі форми)

2)Травень 1996: RFC 1942 (таблиці)

3)Серпень 1996: RFC 1980 (карти зображень на стороні клієнта)

4)Січень 1997: RFC 2070 (інтернаціоналізація)

**HTML 3**

14 січня 1997 р

HTML 3.2 був опублікований як рекомендація W3C. Це була перша версія, розроблена та стандартизована виключно W3C, оскільки IETF закрив свою робочу групу HTML 12 вересня 1996 р[6].

Спочатку з кодовою назвою "Wilbur", HTML 3.2 повністю відкинув математичні формули, узгодив перекриття різних запатентованих розширень і прийняв більшість тегів візуальної розмітки Netscape. Елемент блимання Netscape та елемент Microsoft Marquee були опущені завдяки взаємній домовленості між двома компаніями. Розмітка для математичних формул, подібна до такої в HTML, була стандартизована лише через 14 місяців у MathML[6].

HTML 4

18 грудня 1997 р

HTML 4.0 був опублікований як рекомендація WWW. Він пропонує три варіанти:

1)Строгий, в якому застарілі елементи заборонені.

2)Перехідні, в яких дозволено застарілі елементи.

3)Набір кадрів, в якому дозволено переважно лише елементи, пов’язані з кадрами.

Спочатку з кодовою назвою "Cougar" HTML 4.0 прийняв багато типів елементів та атрибутів, орієнтованих на браузер, але водночас прагнув поступово скасувати функції візуальної розмітки Netscape, позначивши їх як застарілі на користь таблиць стилів. HTML 4 - це програма SGML, що відповідає ISO 8879 - SGML[6].

24 квітня 1998 р

HTML 4.0 був перевиданий з незначними редагуваннями без збільшення номера версії.

24 грудня 1999 р

HTML 4.01 був опублікований як рекомендація WWW. Він пропонує ті самі три варіації, що і HTML 4.0, і остання помилка була опублікована 12 травня 2001 року.

Травень 2000 р

ISO / IEC 15445: 2000 ("ISO HTML", заснований на HTML 4.01 Strict) був опублікований як міжнародний стандарт ISO / IEC. В ISO цей стандарт потрапляє у сферу ISO / IEC JTC1 / SC34 (Спільний технічний комітет ISO / IEC 1, Підкомітет 34 - Мови опису та обробки документів).

Після HTML 4.01 протягом багатьох років не було нової версії HTML, оскільки розробка паралельної мови на основі XML XHTML займала робочу групу W3C HTML на початку та в середині 2000-х[6].

HTML 5

28 жовтня 2014 р

HTML5 був опублікований як рекомендація WWW.

1 листопада 2016 року

HTML 5.1 був опублікований як рекомендація W3C.

14 грудня 2017 року

HTML 5.2 був опублікований як рекомендація WWW.

**2.2 CSS**

CSS розшифровується як каскадні таблиці стилів. Саме мова кодування надає веб-сайту зовнішній вигляд та макет. Поряд з HTML, CSS є фундаментальним для веб-дизайну. Без цього веб-сайти все одно були б звичайним текстом на білому тлі.

До розробки CSS у 1996 році Консорціумом всесвітньої павутини (WWW) веб-сторінки були надзвичайно обмежені як за формою, так і за функціями. Ранні браузери представляли сторінку як гіпертекст - звичайний текст, зображення та посилання на інші гіпертекстові сторінки. Про макет взагалі не можна було говорити, лише абзаци, що пробігали сторінку в одній колонці.

CSS дозволив кілька нововведень у розміщенні веб-сторінок, таких як можливість:

1) Вказувати шрифти, крім стандартних для браузера;

2) Вказувати колір і розмір тексту та посилань;

3)Застосовувати кольори до фонів;

Містять елементи веб-сторінки у вікнах і переміщають ці вікна до певних позицій на сторінці

Вони помістили "стиль" у таблиці стилів, і вперше можна було створити веб-сторінки.

Першим комерційним браузером, який читав і використовував CSS, був Internet Explorer 3 від Microsoft у 1998 році. На сьогоднішній день підтримка певних функцій CSS залежить від браузера. WWW, який досі контролює та створює веб-стандарти, випустив новий стандарт для CSS - CSS3. За допомогою CSS3 розробники сподіваються, що всі основні браузери будуть читати та відображати кожну функцію CSS однаково.

Щоб зрозуміти основи того, як працює CSS, спочатку потрібно трохи зрозуміти сучасний HTML. Веб-розробники викладають сторінки відповідно до "віконної моделі". Веб-сторінка - це ряд вікон, кожен з яких містить окремий елемент. Ці ящики вкладені один в інший.

Наприклад, заголовок сторінки - це поле, і воно містить кілька менших вікон, що містять усі елементи, що складають заголовок: логотип, навігація, кнопки соціальних мереж, кнопки кошика для покупок тощо. За допомогою CSS розробник призначає стилі для " заголовока "поле. У цьому прикладі припустимо, що розробник робить текст всередині заголовка фіолетовим, шрифт Arial і висотою п’ятнадцять пунктів.

Ось де «каскадна» частина каскадних таблиць стилів набуває значення. Стилі шрифтів, застосовані до каскаду заголовка, до всіх елементів, що містяться всередині заголовка. Елементи, що містять текст, такі як навігація, посилання або заклики до дії, будуть фіолетовим, матимуть шрифт Arial та матимуть висоту в п'ятнадцять пунктів.

Один і той самий [HTML](https://uk.wikipedia.org/wiki/HTML) документ може бути відображений по-різному залежно від використаного CSS. Стилі для відображення сторінки можуть бути такі:

1) Стилі автора (інформація подана автором сторінки):

* зовнішні таблиці стилів, зазвичай окремий файл або файли .css;
* внутрішні таблиці стилів, під’єднані всередині документу або блоку;
* стилі для окремого елементу.

2) Стилі користувача

* локальний .css-файл, вказаний користувачем для використання на веб-сторінках і вказаний в налаштуваннях браузера (наприклад [Opera](https://uk.wikipedia.org/wiki/Opera)). [7].

3) Стилі переглядача ([браузера](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80))

* Стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, реалізується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями. [7].

Застосування CSS надає наступні переваги:

#### 1. Спрощує користувальницький інтерфейс

Систематизує оформлення.

#### 2. Робить веб-сторінку більш привабливою для користувача

Мова для каскадного стилю використовується для прикраси веб-сторінки, щоб вона могла виглядати гарно.

#### 3. Налаштовує макет веб-сторінки

Він має різні атрибути, які працюють зі структурою веб-сторінки та роблять її так, як хоче розробник.

#### 4. Легко інтегрується

Існують різні способи впровадження CSS на веб-сторінці, і їх можна використовувати дуже просто.

#### 5. Легке кодування

Ми вже зрозуміли, що CSS – це не мова програмування, а мова стилізації. Існує фіксована та обмежена кількість атрибутів, які потрібно пам’ятати, що робить кодування CSS дійсно дуже простим. [7].

Різниця між версіями CSS:

CSS має різні рівні та профілі. Кожен рівень CSS базується на останньому, як правило, додає нові функції і зазвичай позначається як CSS 1, CSS 2, CSS 3 і CSS 4. Профілі, як правило, є підмножиною одного або декількох рівнів CSS, побудованих для певного пристрою або користувальницького інтерфейсу . В даний час існують профілі для мобільних пристроїв, принтерів та телевізорів

**CSS 1**

Першою специфікацією CSS, яка стала офіційною рекомендацією WWW, є рівень CSS 1, опублікований 17 грудня 1996 р. Håkon Wium Lie і Bert Bos зараховуються як початкові розробники. Його можливості:

Зміна шрифту та наголосу;

Колір тексту, фонів та інших елементів;

Атрибути тексту, такі як інтервал між словами, літерами та рядками тексту;

Вирівнювання тексту, зображень, таблиць та інших елементів;

Поля, межі, відступи та позиціонування для більшості елементів;

Унікальна ідентифікація та загальна класифікація груп атрибутів;

WWW більше не підтримує Рекомендації CSS 1.

**CSS 2**

Специфікація рівня CSS 2 була розроблена WWW і опублікована як рекомендація в травні 1998 року. Набір CSS 1, CSS 2 включає ряд нових можливостей, таких як абсолютне, відносне та фіксоване позиціонування елементів та z-індекс, концепція типи носіїв, підтримка звукових таблиць стилів (які згодом були замінені мовними модулями CSS 3) та двонаправлений текст, а також нові властивості шрифту, такі як тіні. W3C більше не підтримує рекомендації CSS 2.

**CSS 2.1**

Редакція 1 рівня CSS 2, яку часто називають "CSS 2.1", виправляє помилки в CSS 2, видаляє погано підтримувані або не повністю сумісні функції та додає до специфікації вже реалізовані розширення браузера. Щоб відповідати процесу WWW щодо стандартизації технічних специфікацій, CSS 2.1 протягом багатьох років рухався між статусом робочого проекту та статусом рекомендації кандидата.

CSS 2.1 перейшов до запропонованої рекомендації 12 квітня 2011 р. Після розгляду Консультативним комітетом WWW він нарешті був опублікований як Рекомендація WWW 7 червня 2011 р.

CSS 2.1 планувався як перший та остаточний перегляд рівня 2, але робота з низьким пріоритетом над CSS 2.2 розпочалася в 2015 році.

**CSS 3**

На відміну від CSS 2, який є великою єдиною специфікацією, що визначає різні функції, CSS 3 розділений на кілька окремих документів, які називаються "модулями". Кожен модуль додає нові можливості або розширює функції, визначені в CSS 2, зберігаючи зворотну сумісність. Робота над рівнем CSS 3 розпочалася приблизно з моменту публікації оригінальної рекомендації CSS 2. Найбільш ранні проекти CSS 3 були опубліковані в червні 1999 р.

**2.3 PHP**

PHP - це мова сценаріїв загального призначення, особливо придатна для веб-розробки. Спочатку він був створений датсько-канадським програмістом Расмусом Лердорфом у 1994 році . Наразі реалізація PHP виробляється The PHP Group. PHP спочатку розшифровувався як Персональна Домашня Сторінка , але тепер це означає рекурсивний ініціалізм PHP: Гіпертекстовий Препроцесор[8].

Код PHP зазвичай обробляється на веб-сервері інтерпретатором PHP, реалізованим як модуль, демон або як виконуваний файл CGI (Common Gateway Interface). На веб-сервері результат інтерпретованого та виконаного PHP-коду - який може бути будь-яким типом даних, наприклад, згенерованими HTML або даними двійкових зображень - буде формувати цілу або частину відповіді HTTP. Існують різні системи веб-шаблонів, системи управління веб-контентом та веб-фреймворки, які можуть бути використані для організації або сприяння генерації такої відповіді. Крім того, PHP може використовуватися для багатьох завдань програмування поза веб-контекстом, таких як автономні графічні програми та роботизована система управління безпілотниками. Код PHP також може бути безпосередньо виконаний з командного рядка.[8].

Стандартний інтерпретатор PHP, що працює на базі Zend Engine, - це безкоштовне програмне забезпечення, випущене під ліцензією PHP. PHP широко переноситься, і його можна безкоштовно розгорнути на більшості веб-серверів майже на будь-якій операційній системі та платформі.

Мова PHP еволюціонувала без письмової формальної специфікації або стандарту до 2014 року, при цьому оригінальна реалізація виступала як фактичний стандарт, якому інші реалізації мали на меті слідувати. З 2014 року триває робота над створенням офіційної специфікації PHP[8].

Різниця між версіями PHP:

Автором першої версії PHP є Расмус Лердорф. Він був створений у 1994 році як набір сценаріїв Perl з метою моніторингу користувачів Інтернету, які відвідують веб-сайт та переглядають резюме Лердорфа. Незабаром виявилося, що занадто великий обсяг трафіку вимагає відповіді програміста, тому в подальших роботах він використав мову C та додав нові опції. Інтерес посилився, і люди попросили мати можливість використовувати ці інструменти у власних цілях, що призвело до випуску Інструментів особистої домашньої сторінки в 1995 році. Однак Расмус Лердорф не уповільнив темпи - він додав послугу mSQL до PHP та поєднав його з Form Interpreter. Таким чином PHP / FI побачив світ, і в 1997 році була випущена його версія 2.0, яка вже мала кілька тисяч користувачів і підтримувала близько 50 тисяч доменів. З тих пір PHP / FI почали динамічно розвиватися[8].

**PHP 3**

У 1997 р. Ізраїльські програмісти Зеєв Сураскі та Анді Гутманс вирішили продовжити роботи над мовою PHP. Вони виявили, що PHP / FI недостатньо добре справляється з вимогами програми електронної комерції, яку вони розробляли на той час. Тож вони вирішили переписати код з нуля, розробили нову архітектуру, а в 1998 році анонсували нову версію.

У PHP 3 з'явилися основи об'єктно-орієнтованого програмування, але найбільш відмінною рисою була його модульність, тобто можливість збагачення функціональності PHP шляхом додавання нових модулів користувачами. Крім того, PHP 3 почав підтримувати різні бази даних: mSQL, MySQL, PostgreSQL. На думку багатьох програмістів, з вбудованими елементами було набагато простіше обробляти, ніж намагатися створити окремі файли HTML та CGI. Все це зробило загальну ефективність веб-сайтів значно оптимізованою, а управління великими веб-сайтами стало більш ефективним, оскільки компоненти веб-сайтів були зібрані в одному файлі HTML.

**PHP 4**

Однак вищезазначена модульність та питання ефективності постійно призводили до помилок Сураскі та Гутмана. Рухомі прагненням до постійного вдосконалення, вони взялися за подальші роботи над вихідним кодом.

Таким чином, в 1999 році офіційно вийшов Zend Engine - скриптовий механізм, який став основою для розробки PHP 4, який був запущений роком пізніше. Протягом наступних чотирьох років було випущено ще три видання: 4.1, 4.2, 4.3, і кожне з них характеризувалося покращеною безпекою та швидкістю роботи. Говорячи про PHP 4, безсумнівно, слід наголосити на згаданому Zend Engine. Цей механізм сценаріїв із відкритим кодом, який інтерпретує мову програмування PHP, написаний на мові C як оптимізований модульний сервер. Тут важливо відзначити той факт, що вперше він може бути використаний в додатках поза PHP. Механізм управляє пам'яттю та ресурсами, а його продуктивність та розширюваність мали прямий вплив на розробку наступних версій PHP[8].

**PHP 5**

Говорячи про версію 5, слід сказати: "quid pro quo". Нова розроблена об'єктно-орієнтована модель програмування коштувала певної сумісності з попередніми версіями PHP. Звичайно, це не зовсім недолік PHP 5, оскільки модулі були змінені, включаючи модулі, що використовуються для обробки XML та зв'язку з базою даних.

Іншими суттєвими змінами були: реалізація XML-функцій у ядрі програми, повний перепис розширення MySQL - однієї з найпростіших, безкоштовних програм баз даних з відкритим кодом, ефективність яких у поєднанні з PHP 4 не була великою, і в PHP 5 розширення MySQL було оптимізовано для сумісності з MySQL 5.0. Також існувала функція \_\_autoload (), яка була розроблена для прискорення часу кодування та підвищення продуктивності веб-сайту. Також було вдосконалено роботу PHP в системі Windows[8].

У 2009 р. Вийшов PHP 5.3, але вже в 2005 р. З'явилася інформація, що працює над новою версією[8].

**PHP 6**

Unicode. Ключове слово, говорячи про версію 6. Метою було видалити застарілі рішення, ще з часів PHP / FI та PHP3, для усунення постійних проблем із сумісністю. Ці зміни були пов’язані із впровадженням удосконалень об’єктної моделі мовою Unicode.

Творці хотіли додати рідну підтримку Unicode до PHP. Однак проблема, з якою вони зіткнулися, полягала в нестачі програмістів, які бачили необхідність цих змін, а також у проблемах продуктивності в контексті самого перетворення UTF-16[8].

Нарешті, у 2010 році від проекту в його нинішній формі було офіційно відмовлено, і випуск PHP 5.4 був підготовлений без урахування функцій Unicode PHP 6.

Ніколи не випущена версія 6 вважається провалом. Тож не дивно, що ініціативою щодо створення наступної була ініціатива.

**PHP 7**

2014 рік вважається офіційним початком робіт над PHP 7. Основою версії стала експериментальна гілка - PHPNG (PHP Next Generation), яка була розроблена для оптимізації продуктивності[8].

Розробники дуже багато працювали над переробкою коду з метою зменшення споживання пам'яті та підвищення зазначеної продуктивності. Вони впевнені, що досягли своєї мети. Тести для PHP 7 показують вдвічі більшу швидкість, ніж PHP 5,6, і на 50% менше споживання пам'яті. Це безсумнівний успіх цієї версії. Зменшення споживання пам’яті дозволяє покращити обробку запитів, а також можливість побудови мікросервісів на основі PHP. Внутрішні зміни також створюють можливості для майбутніх оптимізацій, які можуть ще більше підвищити ефективність. На цьому етапі варто наголосити на компіляторі JIT, який, у свою чергу, з’явиться у всій красі в найновішому PHP[8].

**2.4 JS**

Що таке JavaScript і чому це важливо?

JavaScript - це мова програмування, що використовується переважно веб-браузерами для створення динамічного та інтерактивного досвіду для користувача. Більшість функцій та додатків, які роблять Інтернет необхідним для сучасного життя, кодуються у певній формі JavaScript.

Найбільш ранні втілення JavaScript були розроблені наприкінці 1990-х для веб-браузера Netscape Navigator. У той час веб-сторінки були статичними, пропонуючи мало взаємодії з користувачами, окрім клацання посилань та завантаження нових сторінок. Вперше JavaScript активував анімацію, адаптивний вміст та перевірку форми на сторінці.

Протягом багатьох років JavaScript функціонував лише на обмеженій кількості браузерів. Microsoft Internet Explorer, найбільша база браузерів, підтримувала JavaScript лише набагато пізніше. Натомість Microsoft створила власний скрипт на стороні клієнта під назвою JScript. У перші дні веб-розробки програмісти, які бажали створювати динамічні веб-сайти, часто були змушені вибирати одну сім'ю браузерів серед інших. Це було менш ніж ідеально, оскільки робило Інтернет менш загальнодоступним.

JavaScript не став стандартизованим і широко прийнятим до 1999 року. Навіть після стандартизації сумісність браузерів залишалася проблемою протягом десяти років.[11].

**Як працює JavaScipt?**

JavaScript - це те, що називається сценарієм на стороні клієнта. Більшість веб-програм, таких як пошукова система, працюють завдяки взаємодії між пристроєм користувача (наприклад, комп’ютером, телефоном або планшетом) та віддаленим сервером. Програмне забезпечення на віддаленому сервері надсилає інформацію клієнту (тобто машині користувача), а програмне забезпечення на стороні клієнта зчитує інформацію та відображає веб-сторінку на екрані.

Клієнтський сценарій - це мова програмування, яка повністю виконує свої завдання на машині клієнта, і для її функціонування не потрібно взаємодіяти з сервером. Наприклад, якщо на вашому комп’ютері завантажена веб-сторінка, а постачальник послуг Інтернету не працює, ви все ще можете взаємодіяти з веб-сторінками, вже завантаженими у ваш браузер. Однак ви не зможете переходити на нові веб-сторінки або отримувати доступ до будь-яких даних, розташованих віддалено.

Деякі з динамічних удосконалень веб-сайтів, які виконуються JavaScript:

1) Автозаповнення.

2)Завантаження нового вмісту або даних на сторінку без перезавантаження сторінки.

3)Ефекти перекидання та випадаючі меню.

4)Анімація елементів сторінки, таких як вицвітання, зміна розміру або переміщення.

5)Відтворення аудіо та відео.

6)Перевірка введення з веб-форм.

7)Виправлення проблем із сумісністю браузера.

Хоча JavaScript є мовою на стороні клієнта, деякі найпотужніші функції включають асинхронну взаємодію з віддаленим сервером. Асинхронний просто означає, що JavaScript здатний взаємодіяти з сервером у фоновому режимі, не перериваючи взаємодію користувача, що відбувається на передньому плані.[11].

Візьмемо, наприклад, пошукову систему. Сьогодні майже всі пошукові системи мають функцію автозаповнення. Користувач починає вводити слово у вікно пошуку, а нижче перелік можливих пошукових термінів або фраз. Досвід бездоганний. Запропоновані пошукові терміни з’являються без перезавантаження сторінки.У фоновому режимі JavaScript зчитує літери за типом користувача, відправляє ці листи на віддалений сервер, а сервер надсилає пропозиції назад.

Програмне забезпечення на стороні сервера аналізує слова та запускає алгоритми, щоб передбачити пошуковий термін користувача. Такі програми диявольськи великі та складні. JavaScript на машині клієнта максимально простий і маленький, щоб не сповільнювати взаємодію користувача. Зв'язок між JavaScript та серверною програмою обмежується пропускною здатністю користувача. Ось чому розробники надають пріоритет ефективності функцій JavaScript і роблять обсяг даних, переданих між програмами, якомога меншим.

Лише після того, як користувач вибрав пошуковий термін, вся сторінка перезавантажується та видає результати пошуку. Такі двигуни, як Google, зменшили або усунули необхідність перезавантажувати навіть для цього кроку. Вони просто дають результати, використовуючи той самий асинхронний процес[11].

**Майбутнє JavaScript**

Хоча JavaScript не є єдиною мовою сценаріїв на стороні клієнта в Інтернеті, вона була однією з перших і досі є найбільш широко використовуваною. Багато розробників вважають, що JavaScript неефективний і витончений, тому за ці роки вони зробили багато вдосконалення мови. Підприємливі програмісти створили бібліотеки JavaScript - більш стислі мови, побудовані з будівельних блоків JavaScript, які є менш складними і можуть бути націлені на конкретні програми.

Наприклад, JQuery - це бібліотека JavaScript, яка спрощує та розширює багато анімаційних та інтерактивних функцій JavaScript, тоді як Backbone.js полегшує адаптивний дизайн.

JavaScript став невід’ємною частиною досвіду роботи в Інтернеті, оскільки розробники вбудовують у свої програми більше взаємодії та складності. Пошукові системи, електронна комерція, системи управління вмістом, адаптивний дизайн, соціальні медіа та телефонні додатки без цього були б неможливими.

**2.5 Висновок за розділом 2**

В даному розділі було розглянуто мову розмітки гіпертекстових документів HTML, каскадні таблиці стилів які використовуються для зовнішнього вигляду, процесор гіпертексту PHP також мову програмування JavaScript, що дозволяє зробити Web -сторінку інтерактивною.

**РОЗДІЛ 3. ПРОЕКТНА ЧАСТИНА**

**3.1 Розробка інтерфейсу веб-сайту**

Відповідно до розробленої структури була спроектована головна сторінка сайту (Рис. 3.1). Вона містить всі основні структурні елементи, перехід по яких здійснюється за допомогою гіперпосилань.

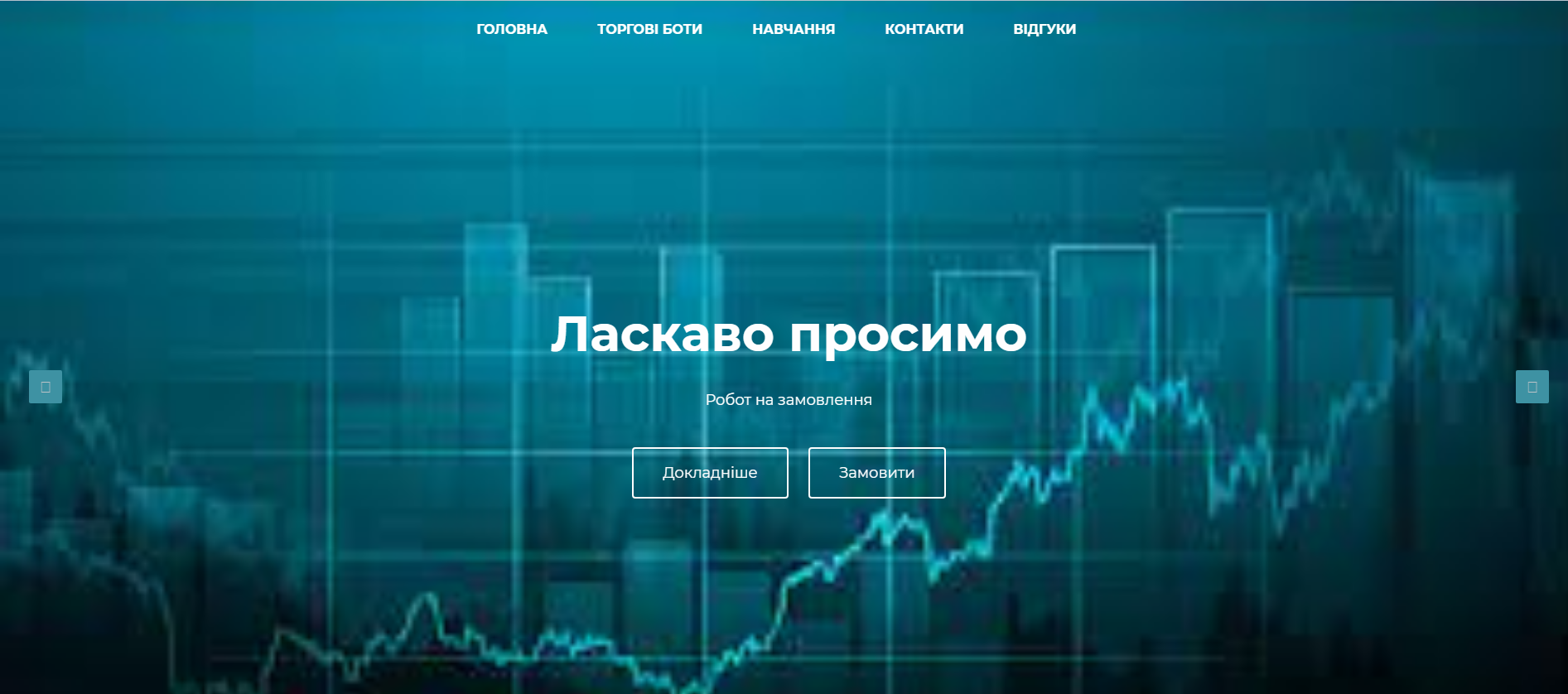


Рисунок 3.1 – Головна сторінка сайту

Опис меню «Головна» Коли заходимо на сайт то першою з’явиться головна сторінка. Головна сторінка складається з шапки, основного меню, і змістовної частини.

В шапці знаходяться головні розділи всього веб-сайту, такі як («Головна», «Торгові боти,» «Навчання», «Контакти», «Відгуки»)

В змістовній частині головної сторінки знаходиться загальна інформація про мету створення сайту.

Зайшовши на сторінку «Торгові боти» ви побачите інформацію про продукт який пропонують на цьому веб-сайті.

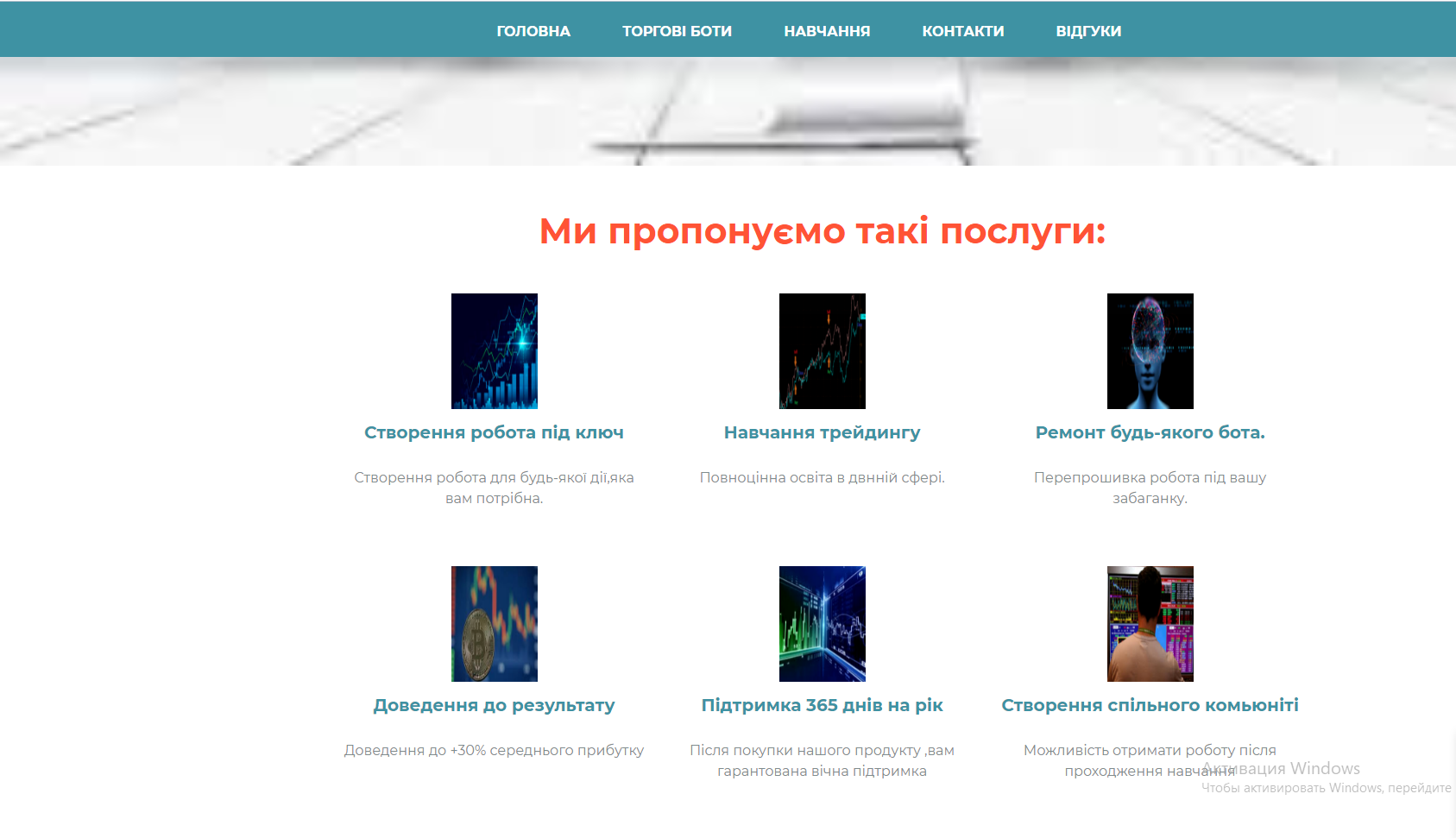


Рисунок 3.2 – Сторінка «Торгові боти»

Зайшовши на сторінку «Навчання» можна побачити основну інформацію про навчання, і залишити свою інформацію за допомогою форми зворотного зв’язку.

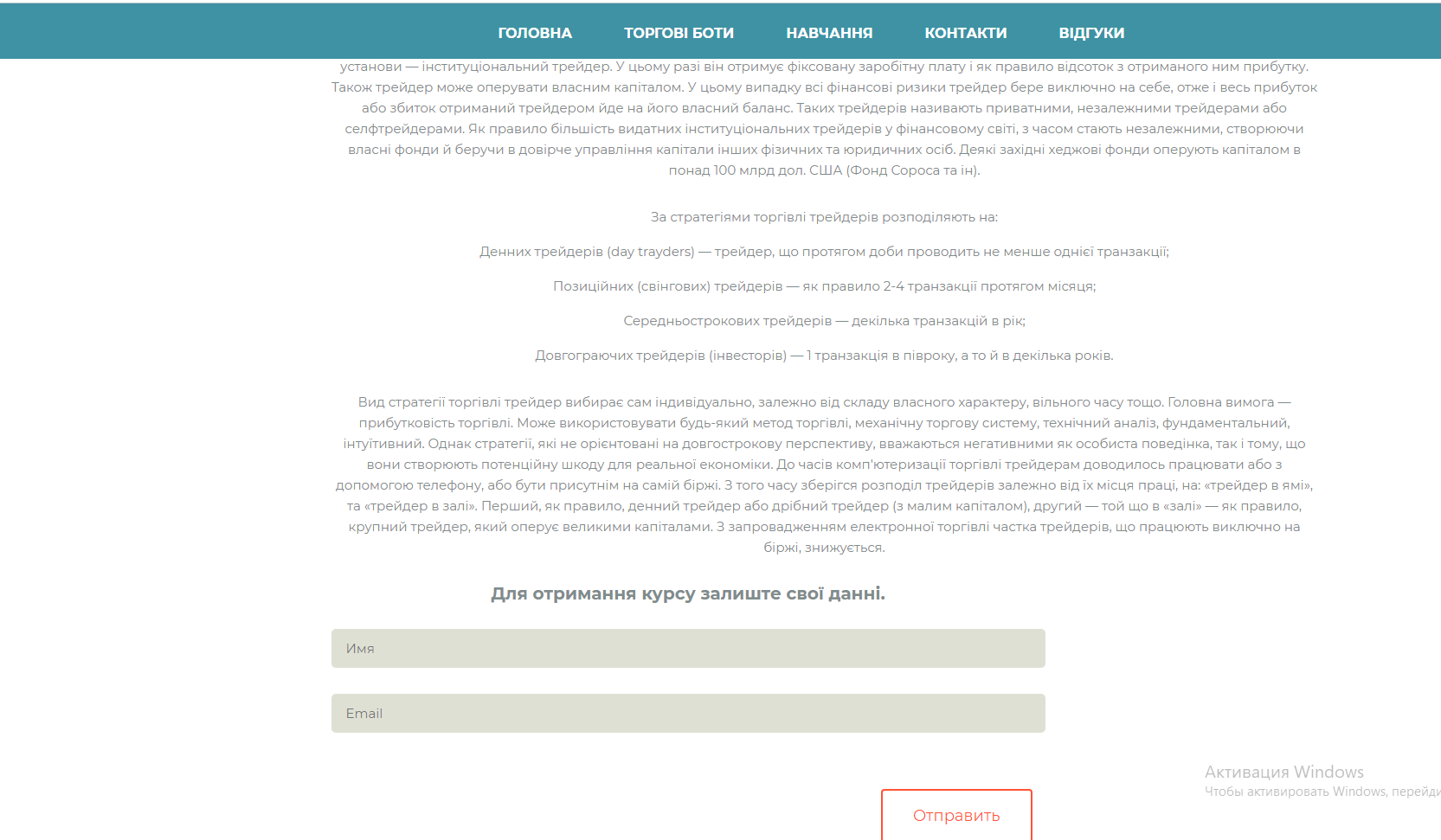


Рисунок 3.3 – Сторінка «Навчання»

Зайшовши на сторінку «Контакти» побачимо основну інформацію про керівників . Також є можливість залишити повідомлення розробникам через форму зворотного зв’язку.

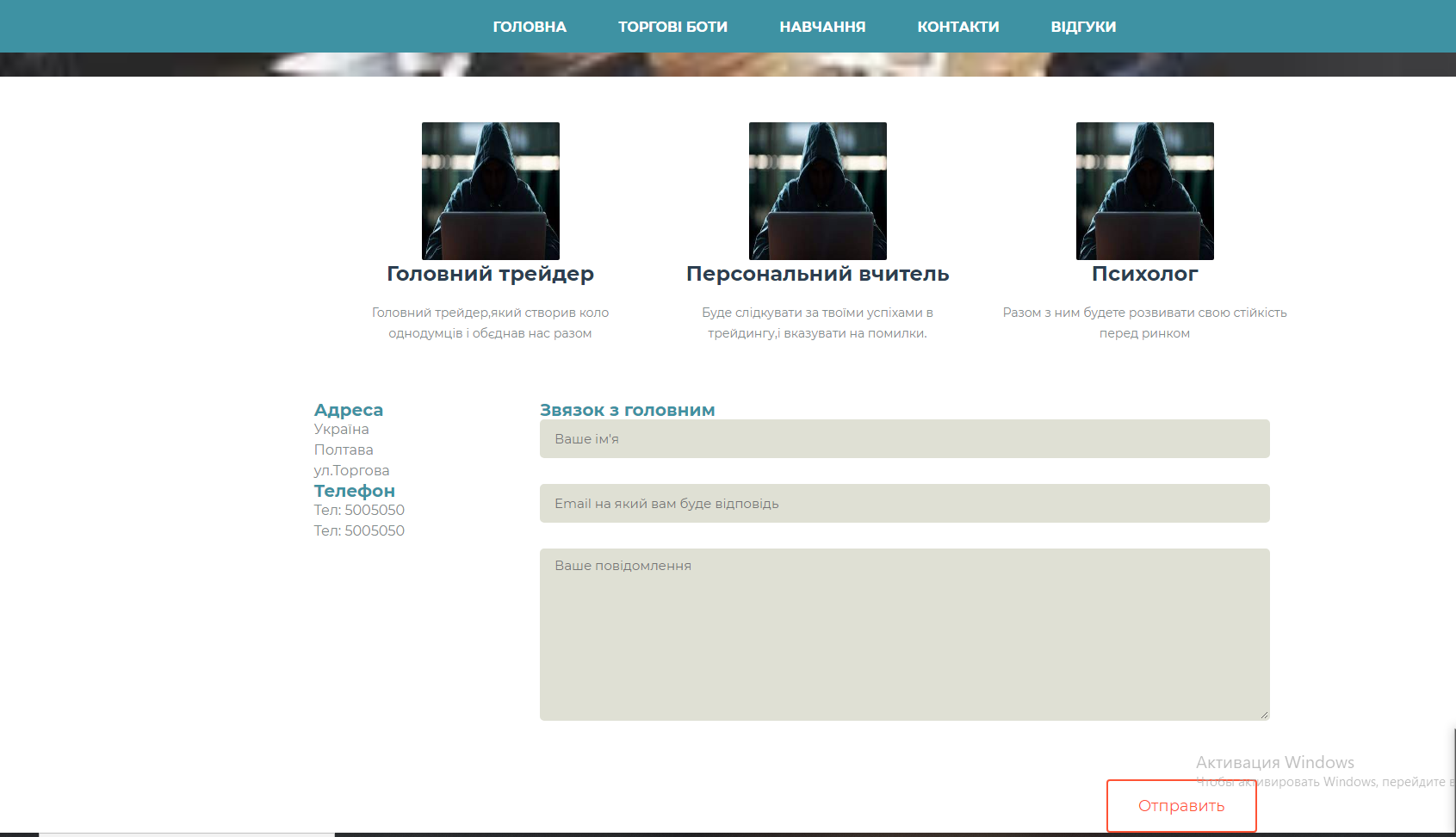


Рисунок 3.4 – Сторінка «Контакти»

В змістовній частині сторінки «Відгуки» можна залишити свій відгук про роботу з нами.

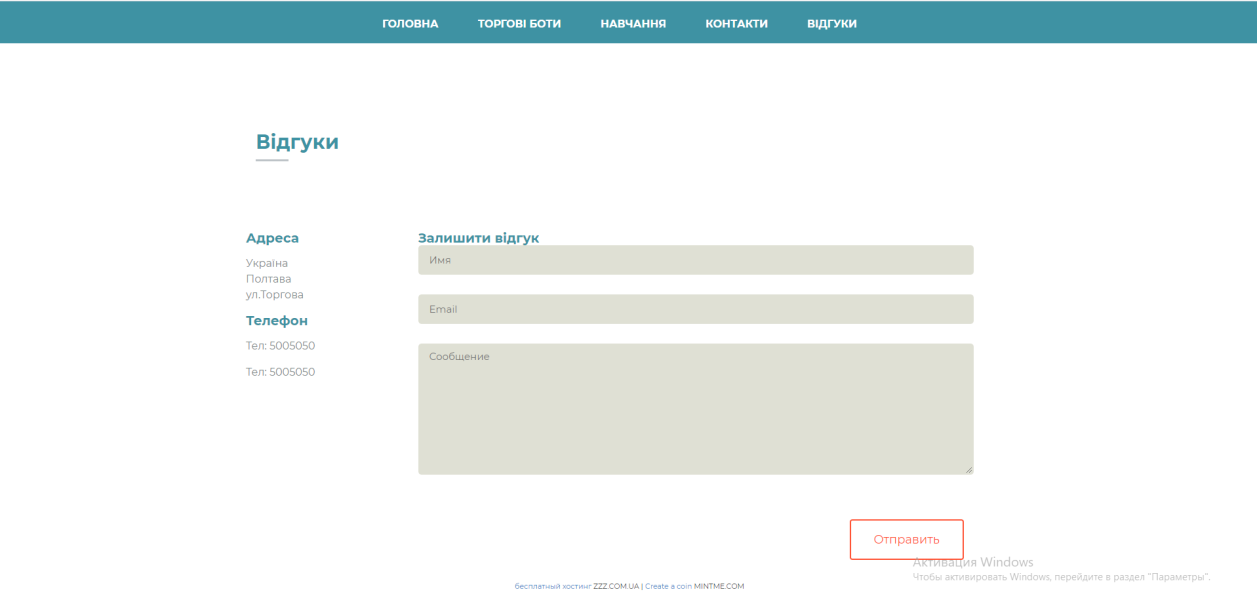


Рисунок 3.5 – Сторінка «Відгуки»

**3.2 Реєстрація сайту в мережі Інтернет**

Хо́стинг – послуга надавання дискового простору, підключення до мережі та інших ресурсів для розміщення фізичної інформації на сервері, що постійно перебуває в мережі (наприклад інтернеті).

Поняття хостингу включає в себе широкий спектр послуг із використанням різного апаратного та програмного забезпечення. Зазвичай під поняттям послуги хостингу мають на увазі, як мінімум, послугу розміщення файлів сайту на сервері, на якому запущене ПЗ, необхідне для обробки запитів до цих файлів (веб-сервер).

Провайдерами хостингу можуть виступати як компанії, що спеціалізуються на цих послугах («хостери»), так і великі провайдери інформаційних послуг, що спеціалізуються на інших послугах (такі як Google, Microsoft, Yahoo та інші).

Розрізняють безкоштовний та платний хостинг. Безкоштовні «хостери» заробляють на тому, що розміщують рекламу на своїх сайтах або на наданні інших платних послуг (у пакеті з безкоштовними або опціонально).

**Види хостингу**

Повнофункціональний хостинг:

* [віртуальний хостинг](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) (virtual hosting або shared hosting) – користувачеві надається частина місця на [диску](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA) для розміщення [вебсайтів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82). При цьому середовище виконання вебсервісів єдине для багатьох користувачів, а апаратні і програмні ресурси розподілені між усіма користувачами на одному [сервері](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80), де може розміщуватись від 50 до 1000 користувачів. Перевагами віртуального хостингу є відносно низькі ціни та набір послуг, що є адекватним для функціонування невеликого та оптимізованого сайту. Недоліками можна вважати те, що через розподілення ресурсів серверу між багатьма користувачами, надмірне споживання цих ресурсів одним сайтом може вплинути на роботу інших.
* [віртуальний виділений сервер](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) (VPS або VDS) – послуга, в рамках якої користувачеві надається так званий віртуальний виділений сервер. Спосіб управління [операційною системою](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) здебільшого відповідає управлінню фізичним виділеним сервером. Зокрема: права адміністратора, root-доступ, власні [IP-адреси](https://uk.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%B0), порти, правила фільтрування і таблиці маршрутизації. Віртуальний сервер надає користувачеві більше можливостей і привілеїв, а часто і більше ресурсів, ніж віртуальний хостинг. Водночас він є в середньому і більш дорогим.
* [виділений сервер](https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) (dedicated server) – надається сервер цілком. Використовується для реалізації нестандартних завдань (сервісів), а також розміщення «важких» вебпроектів, які не можуть співіснувати на одному сервері з іншими проектами і вимагають для себе всі ресурси сервера. Виділений сервер вважається найбільш ефективним, і водночас найбільш дорогим рішенням серед усіх видів хостингу. Управління таким сервером потребує від користувача більше технічних знань і навичок, ніж управління віртуальним хостингом.
* [колокація](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D0%BE%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9_%D1%86%D0%B5%D0%BD%D1%82%D1%80) (collocation) – надання місця в дата центрі провайдера для обладнання клієнта (зазвичай шляхом монтажу в стійці) і підключення його до [Інтернету](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Дає можливість користуватися інфраструктурою дата центру, а саме системами охолодження повітря, пожежної безпеки тощо. Цією послугою користуються переважно досвідчені клієнти, що мають достатньо навичок для підключення і адміністрування власних серверів, або компанії, що самі надають інформаційні послуги.
* хмарний хостинг (cloud hosting або cloud storage) – це послуга з розміщення файлів користувача, за якої дані зберігаються на багатьох серверах, що розподілені у мережі. Файли зберігаються у так званій «хмарі», що фізично складається з серверів, які можуть знаходитися далеко один від іншого, але з точки зору користувача працюють як один потужний віртуальний сервер. Перевагами хмарного хостинга є можливість колективної роботи з даними та відсутність прив'язки до ресурсів одного окремого серверу.
* реселлер хостинг (reseller hosting) – хостинг з послугою перепродажу. Користувачеві надається можливість розподіляти дисковий простір і ресурси свого віртуального хостингу або серверу з метою розміщення на ньому сайтів третіх осіб, що можуть бути його клієнтами. Пакет послуг такого хостингу зазвичай включає спеціальне ПЗ для управління клієнтською базою, ресурсами, що надані клієнтам, тощо.

Мій сайт було підключено до безкоштовного хостингу <http://trading-start.zzz.com.ua/index.html>

**3.3 Тестування веб-сайту в різних браузерах**

Кросбраузерність сайту (від cross-browser) – однакове (або з незначними спотвореннями) відображення та функціонування сайту в різних Інтернет-браузерах. Проблема на сьогоднішній день є непростим завданням, а тому вважається одним з ознак професійного створення сайтів і веб-дизайну. Перед розробником сайту стоїть завдання: він повинен забезпечити максимальну сумісність сайту з різними браузерами, щоб користувач міг з однаковим комфортом переглядати і працювати з сайтом як в Internet Explorer, так і в Opera та інших браузерах.

Тестування сайту на кросбраузерність є обов'язковим завданням, адже відмінності в особливостях кожного з браузерів вельми великі, а розробник не може передбачити, чим користується відвідувач для перегляду сайту.

Серед особливостей сайту можуть бути такі, як різні способи виведення шрифтів або їх розміри, а також можливу відсутність деяких з них на комп'ютері користувача. І це самі незначні проблеми, які можуть виникнути.

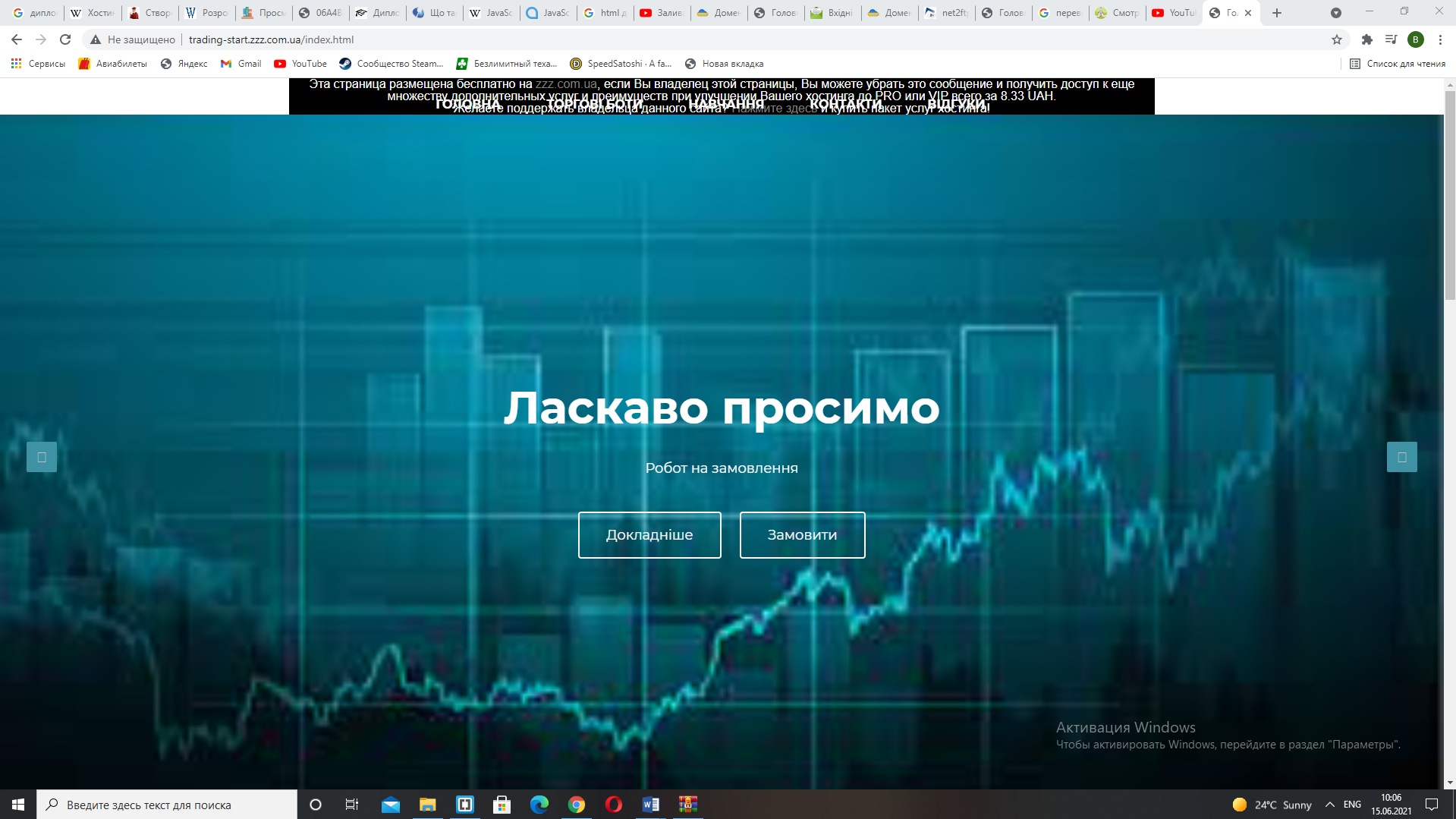


Рисунок 3.6 – браузер Google Chrome

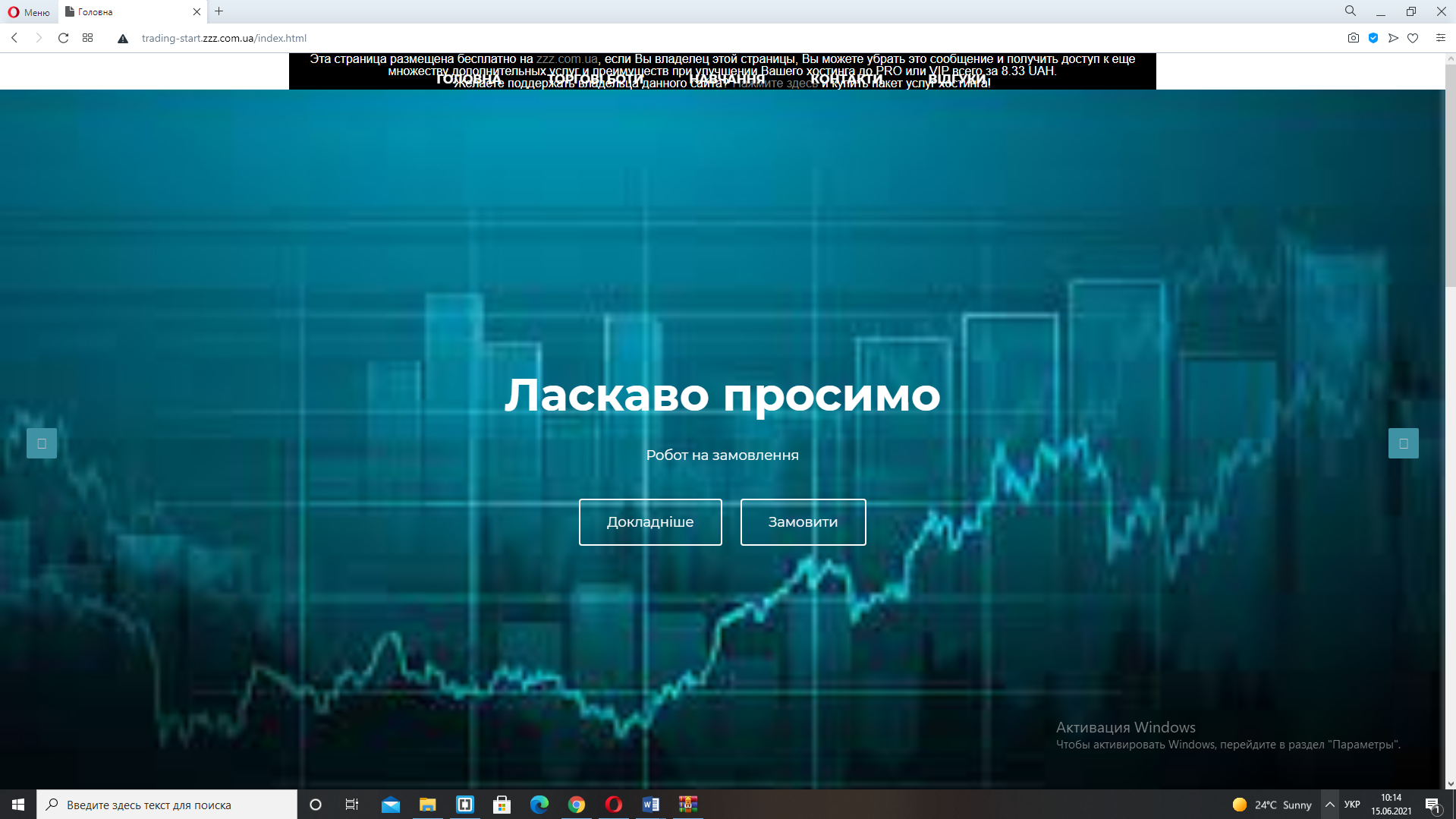


Рисунок 3.7 – браузер Opera

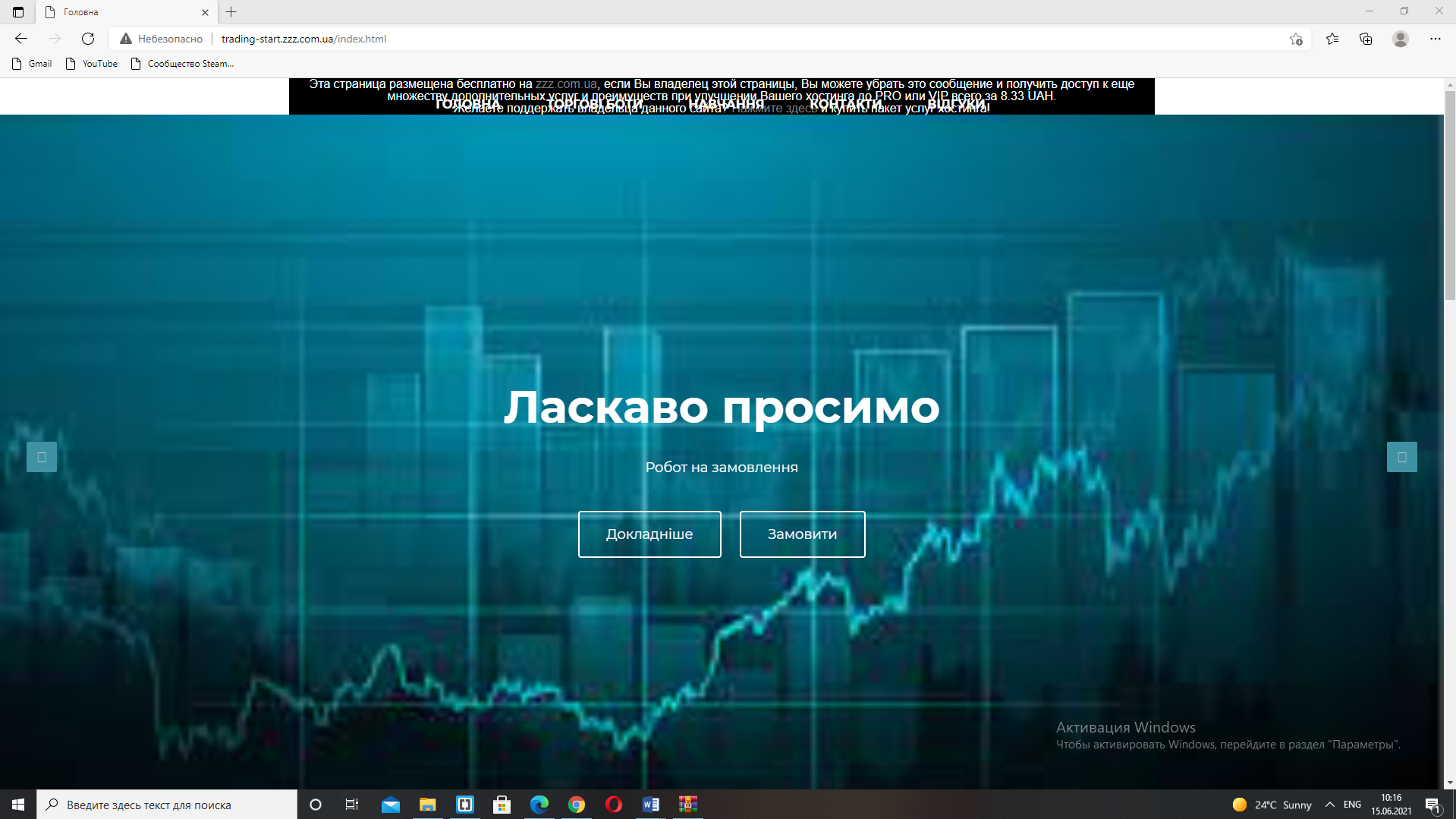


Рисунок 3.8 – браузер Microsoft Edge

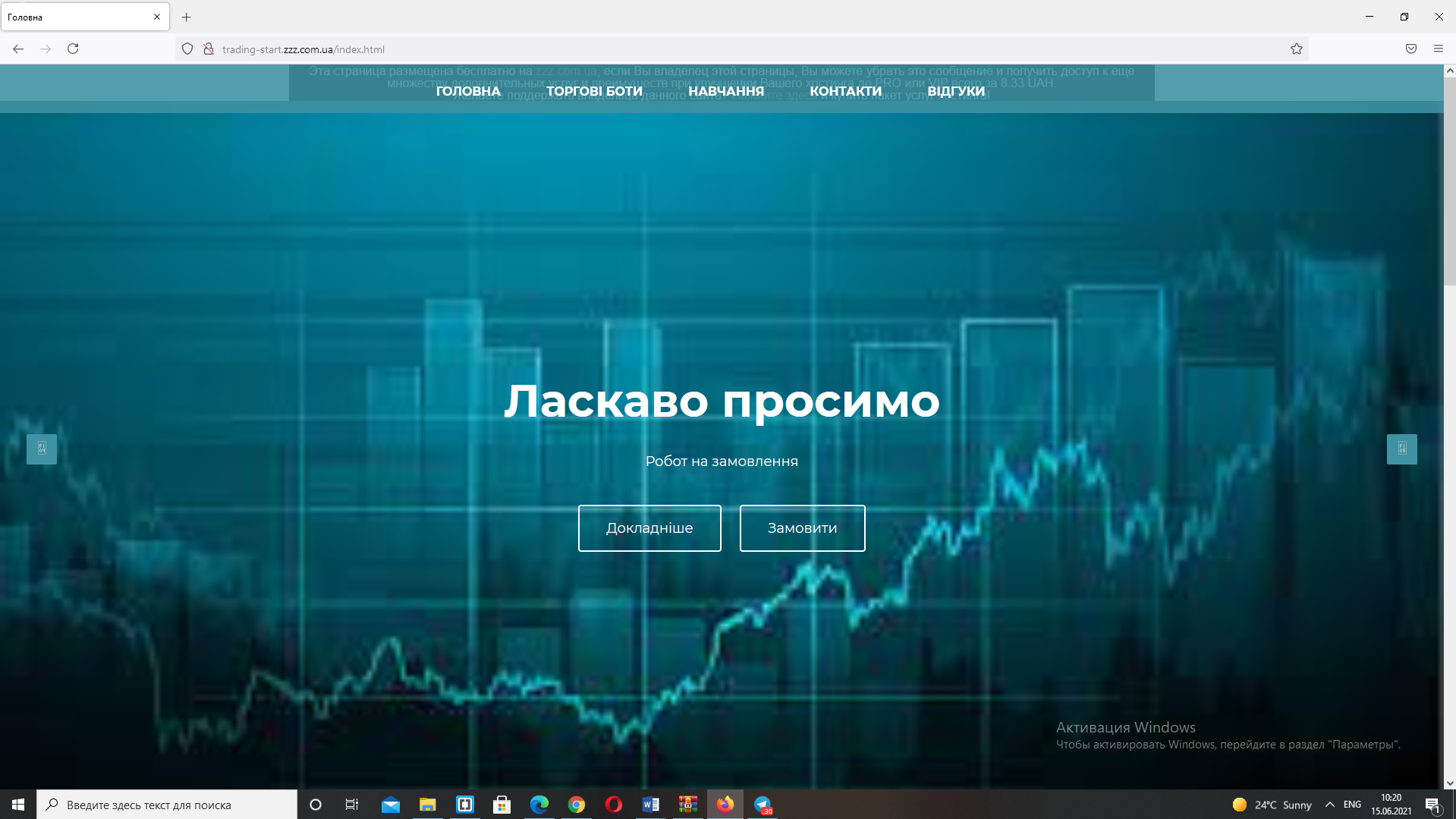


Рисунок 3.9 – браузер Mozila Firefox



Рисунок 3.10 – перегляд сайту зі смартфону

**3.4 Висновки до розділу 3**

Згідно завдання було створено веб-сайт. На головній сторінці розміщено, основне меню сайту (для навігації по його структурі), Також сайт має інформативні сторінки, які мають інформативний вміст для людей і корисну інформацію для користувачів. Сайт має гарну адаптивність, зручний інтерфейс, та яскраве наповнення. Цей сайт можна переглядати на різних пристроях не хвилюючись за те, що основна інформація та зображення змінять своє розташування.

**ВИСНОВОК**

В ході виконання дипломної роботи був отриманий повнофункціональний web-сайт, повністю готовий до застосування. З його допомогою користувачі зможуть отримувати необхідну інформацію та дізнатися щось нове в світі новітніх технологій. При розміщенні його в глобальній мережі географія розповсюдження зростає до масштабів всього світу.

При розробці web-сайту були проаналізовані сучасні web-технології, що дозволяють створювати інтерактивні та адаптивні web-сторінки. Завдання було виконане в редакторі Brackets

Розроблений сайт задовольняє всі вимоги, поставлені на етапі постановки завдання. Як для подальшого вдосконалення web-сайту представляється можлива доробка інтерфейсу сайту з метою подальшого підвищення його інформативності, привабливості і зручності.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1) Основні поняття мови HTML та структура документів [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/vivcaemowebdizajndistancijno/html/lekcia-3-osnovni-ponatta-movi-html-ta-struktura-dokumentiv>.

2) Створення та розміщення власних веб-сайтів на безкоштовних хостінгах. [Електронний ресурс] – Режим доступу до ресурсу: https://infopedia.su/7xcc5d.html.

3) Структура сайтів [Електронний ресурс]. – 11. – Режим доступу до ресурсу: http://webkondratyuk12.blogspot.com/.

4) Поняття, структура та різновиди веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: https://infourok.ru/ponyattya-struktura-ta-riznovidi-vebsaytiv-1316632.html.

5) Хто такі frontend- і backend-розробники? [Електронний ресурс] – Режим доступу до ресурсу: https://te.itstep.org/blog/frontend-and-backend-developers.

6) HTML [Електронний ресурс] – Режим доступу до ресурсу: https://www.wigi.wiki/wiki/uk/HTML..

7) Опис інформаційного забезпечення системи для розробки сайту [Електронний ресурс] – Режим доступу до ресурсу: <https://lektsii.org/3-118922.html>.

8) PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/PHP>.

9) Мова програмування PHP [Електронний ресурс] – Режим доступу до ресурсу: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html>.

10) ВЕРСІЇ CSS ПРО ПЕРЕВАГИ CSS І ТРОХИ ІСТОРІЇ [Електронний ресурс] – Режим доступу до ресурсу: <http://ruszura.in.ua/uncategorized/versiji-css-pro-perevahy-css-i-trohy-istoriji.html>.

11) Мова JavaScript та її можливості [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti.

**ДОДАТОК А**

index.html

<!DOCTYPE html>

<html class="no-js">

<head>

<title>Головна</title>

<meta charset="utf-8">

<meta class="viewport" name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<link href='http://fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>

<link href='css/font-awesome.min.css' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="css/bootstrap.css">

<link rel="stylesheet" href="css/animate.min.css">

<link rel="stylesheet" href="css/owl.theme.css">

<link rel="stylesheet" href="css/owl.carousel.css">

<link rel="stylesheet" href="css/superslides.css">

<link rel="stylesheet" href="css/magnific-popup.css">

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div id="preload">

<div class="spinner">

<div class="rect1"></div>

<div class="rect2"></div>

<div class="rect3"></div>

<div class="rect4"></div>

<div class="rect5"></div>

</div>

</div>

<nav class="navbar navbar-fixed-top navbar-custom" role="navigation">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#nav">

<span class="sr-only"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<div class="collapse navbar-collapse" id="nav">

<ul class="nav navbar-nav uppercase">

<li class="links"><a class="scroll" href="index.html">Головна</a></li>

<li class="links"><a class="scroll" href="Tradebot.html">Торгові боти</a></li>

<li class="links"><a class="scroll" href="teaching.html">Навчання</a></li>

<li class="links"><a class="scroll" href="Contacts.html" >Контакти</a>

<li class="links"><a class="scroll" href="response.html">Відгуки</a></li>

</ul>

</div>

</div>

</nav>

<div id="slides">

<ul class="slides-container ">

<!--Slide 1-->

<li>

<img src="img/slide1.jpg" alt="">

<div class="container text-center">

<h2 class="slide-one wow fadeInDown">Ласкаво просимо</h2>

<p class="slide-one">Робот на замовлення</p>

<a class="button button-light-red wow fadeInLeft" href="service.html">Докладніше</a>

<a class="button button-light-red wow fadeInRight" href="service.html">Замовити</a>

</div>

</li>

<li>

<img src="img/slide2.jpg" alt="">

<div class="container text-center">

<h2 class="slide-two wow fadeInDown">Ласкаво просимо</h2>

<p class="slide-two">Індивідульна розробка</p>

<a class="button button-yellow wow fadeInLeft" href="service.html">Докладніше</a>

<a class="button button-yellow wow fadeInRight" href="service.html">Замовити</a>

</div>

</li>

<li>

<img src="img/slide3.jpg" alt="">

<div class="container text-center">

<h2 class="slide-three wow fadeInDown">Ласкаво просимо</h2>

<p class="slide-three">Підтримка 24/7</p>

<a class="button button-white wow fadeInLeft" href="service.html">Докладніше</a>

<a class="button button-white wow fadeInRight" href="service.html">Замовити</a>

</div>

</li>

</ul>

<nav class="slides-navigation">

<div class="container">

<a href="#" class="next"><i class="fa fa-angle-right"></i></a>

<a href="#" class="prev"><i class="fa fa-angle-left"></i></a>

</div>

</nav>

</div>

<div id="home">

<div class="container">

<div class="row">

<div class="col-lg-12 text-center">

<!--Header-->

<h1 class="wow fadeInUp">Наші послуги</h1>

</div>

<div class="col-lg-12 home-content text-center">

<div class="row">

<div class="col-lg-4 wow fadeInLeft">

<h3>Торговий бот під ключ</h3>

<p class="detail">Опис торгових ботів</p>

<a class="button button-light-red my-red" href="service.html">Докладніше</a>

</div>

<div class="col-lg-4 wow fadeInUp"><img src="img/diamond.png" alt=""></div>

<div class="col-lg-4 wow fadeInRight">

<h3>Навчання</h3>

<p class="detail">Доведення до результату</p>

<a class="button button-light-red my-red" href="about.html">Докладніше</a>

</div>

</div>

</div>

</div>

</div>

</div>

<div id="portfolio">

<div class="container">

<div class="row">

<div class="col-lg-12 col-xs-12 text-center">

<div class="row">

<!--Header-->

<h1 class="uppercase wow fadeInDown">Торгові боти</h1>

<!--Sub Head-->

<h2 class="wow fadeInUp">Автоматизація торгівлі<br> для полегшення роботи трейдера.</h2>

<div class="line"></div>

<!--Image-->

<div class="browser clearfix">

<!--Left Image-->

<div class="left pull-left wow fadeInLeft"><img class="img-responsive" src="img/browserleft.jpg" alt=""></div>

<!--Center Image-->

<div class="center wow fadeInUp"><img class="img-responsive" src="img/browser.jpg" alt=""></div>

<!--Right Image-->

<div class="right pull-right clearfix wow fadeInRight"><img class="img-responsive" src="img/browserright.jpg" alt=""></div>

</div>

</div>

</div>

</div>

</div>

</div>

<div id="content">

<div class="container">

<div class="row">

<div class="col-lg-12 contents">

<div class="row">

<div class="col-lg-6 box">

<ul>

<li class="wow fadeInLeft">

<img src="img/ico-img-1.png" alt="">

<div class="box">

<h3>Створення під ключ</h3>

<p>Створення робота для будь-яких робіт під будь-яку платформу.</p>

</div>

</li>

<li class="boxes wow fadeInLeft">

<img src="img/ico-img-2.png" alt="">

<div class="box">

<h3>Автоматичне налаштування</h3>

<p>Робот встановить свої налаштування автоматично під вибрану платформу.</p>

</div>

</li>

<li class="boxes wow fadeInLeft">

<img src="img/ico-img-1.png" alt="">

<div class="box">

<h3>Легкий в розумінні</h3>

<p>Легкість з налаштуванняі і вирішенням проблем</p>

</div>

</li>

<li class="boxes wow fadeInLeft">

<img src="img/ico-img-2.png" alt="">

<div class="box">

<h3>Доступна ціна на ринку</h3>

<p>Ціна не перевищуе аналогічних роботів на ринку</p>

</div>

</li>

</ul>

</div>

<!--Video-->

<div class="col-lg-6 col-xs-12">

<div class="video wow fadeInRight">

<iframe width="400" height="225" video src="https://www.youtube.com/watch?v=DfHLBK36Qtg" frameborder="0" allowfullscreen></iframe>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<script src="js/jquery-2.1.3.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/jquery.superslides.min.js"></script>

<script type="text/javascript" script>

<script src="js/retina-1.1.0.min.js"></script>

<script src="js/wow.min.js"></script>

<script src="js/jquery.fitvids.js"></script>

<script src="js/smooth-scroll.js"></script>

<script src="js/jquery.magnific-popup.js"></script>

<script src="js/main.js"></script>

</body>

</html>